Computers Math. Applic. Vol. 21, No. 5, pp. 137–144, 1991
Printed in Great Britain.

0097-4943/91 $3.00 + 0.00

AN ITERATIVE METHOD FOR SYSTEMS OF NONLINEAR HYPERBOLIC EQUATIONS

JEFFREY S. SCROGGS
ICASE, NASA Langley Research Center,
Hampton, Virginia

Abstract—An iterative algorithm for the efficient solution of systems of nonlinear hyperbolic equations is presented. Parallelism is evident at several levels. In the formation of the iteration, the equations are decoupled, thereby providing large grain parallelism. Parallelism may also be exploited within the solves for each equation. Convergence of the iteration is established via a bounding function argument. Experimental results in two-dimensions are presented.

1. INTRODUCTION

An iterative algorithm suitable for the solution of a system of nonlinear hyperbolic partial differential equations in multiple dimensions is discussed. Convergence is established analytically in the continuous case, and demonstrated numerically in the discrete case.

Many numerical methods currently available for solving systems of nonlinear PDEs have limited parallelism due to strong coupling between the equations (for example, Newton's method requires the computation of a Jacobi matrix). The method presented here decouples the PDEs by linearizing the convection coefficient for a space-time domain. Decoupling the equations provides opportunity to exploit large-grain parallelism. More parallelism may be exploited when solving each decoupled equation. The parallelism available in this method makes it well suited for the machines which represent the state-of-the-art for numerical computations. The method is also simple to implement, making the method a viable alternative to other more complicated methods for nonlinear equations.

The iteration is presented in Section 3. Convergence of the method is established analytically in Section 4, and numerical experiments demonstrate that the solution improves by more than a digit in each iteration in Section 6. This method is an extension of an iterative method for one-dimensional scalar equations [1,2] to systems of equations in multiple dimensions.

2. PROBLEM

Consider the following initial-boundary value problem

$$\frac{\partial U}{\partial t} + \sum_{i=1}^{n} F_i(t,x,U)\frac{\partial U}{\partial x_i} + R(t,x,U)U = 0, \tag{1}$$

where $U = (u_1, u_2, \ldots, u_m)$ is a vector of $m \geq 1$ components. Assume that the spatial domain can be specified by a real valued function $d(x)$ as $\Pi = \{x | d(x) < 0\}$. The range of the temporal variable is $0 < t < T$, and the complete domain is denoted $\Omega = (0,T) \times \Pi = \{(t,x) | \Delta(t,x) < 0\}$, where $\Delta(x,t) = td(x)$. The coefficients R and F_i, for $i = 1$ to n, are matrix-valued functions in $\mathbf{R}^{m\times n}$.

Denote by $F_{i,j}$ the j^{th} row of the matrix F_i, and denote the l^{th} component of $F_{i,j}$ by $f_{i,j,l}$. Similarly, $r_{j,l}$ denotes the component of R that is in the j^{th} row and l^{th} column.

Research conducted at ICASE, NASA Langley Research Center, Hampton, Virginia, supported by NASA Contract No. NAS1-18605.

Typeset by $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX

The data

$$u_j(t,x) = \alpha_j(t,x) \tag{2}$$

serve as initial guess when $t = 0$ and $x \in \Pi$, and as the boundary data for the portion of $\partial\Pi$ such that

$$\psi_j(t,x,U) := F_{i,j}(t,x,U) \bullet \operatorname{grad}(\Delta) < 0. \tag{3}$$

Here, grad(Δ) is the gradient of Δ with respect to the spatial variables. This is analogous to the inflow boundary conditions of fluid dynamics. Other types of boundary conditions may be possible [3], but are not studied here.

3. ITERATION

From equation (1), each component of the solution satisfies

$$\begin{aligned}\frac{\partial u_j}{\partial t} + \sum_i^n f_{i,j,j}(t,x,U)\frac{\partial u_j}{\partial x_i} + r_{j,j}(t,x,U)u_j = \\ -\sum_i^n \sum_{l \neq j}^m f_{i,j,l}(t,x,U)\frac{\partial u_l}{\partial x_i} - \sum_{l\neq j}^m r_{j,l}(t,x,U)u_l.\end{aligned} \tag{4}$$

Let $U^k = (u_1^k, \ldots, u_n^k)$ be the k^{th} iterate. The equation for the j^{th} component is linearized by using the most recent iterate only in the places where the j^{th} component is differentiated. This results in the equation

$$\begin{aligned}\frac{\partial u_j^{k+1}}{\partial t} + \sum_i^n f_{i,j,j}(t,x,U^k)\frac{\partial u_j^{k+1}}{\partial x_i} + r_{j,j}(t,x,U^k)u_j^{k+1} = \\ -\sum_i^n \sum_{l \neq j}^m f_{i,j,l}(t,x,U^k)\frac{\partial u_l^k}{\partial x_i} - \sum_{l\neq j}^m r_{j,l}(t,x,U^k)u_l^k.\end{aligned} \tag{5}$$

Thus, the equations for each of the iterates have been decoupled. This is analogous to the Jacobi iteration for systems of linear equations. Each u_j^k satisfies (2) for the portion of $\partial\Omega$ such that $\psi_j(t,x,U^{k-1}) < 0$.

Forms of the iteration other than (7) are possible and may be desirable. For example, if parallelism is not important, then it may be advantageous to linearize but not decouple the equations.

4. CONVERGENCE

The convergence of the iteration has its roots in the existence and uniqueness proofs of partial differential equations [4–6]. However, existence and uniqueness are not the topics of this paper. The iteration defined by equation (7) is shown to be a contraction provided the domain is restricted to regions where the solution is suficiently smooth and continuous. The numerical experiments of Section 6 demonstrate that the restriction on the smoothness of U is only for the sake of the analysis, and the iteration will converge under less strict conditions. In the statement of the theorem, $\bar{\Omega}$ is the closure of Ω, and

$$\|h(t,x)\|_{\hat{T}} = \left[\int_0^{\hat{T}} \int_{\Pi} (h(t,x))^2 dx dt\right]^{1/2}$$

denotes the L_2 norm.

THEOREM 1. *Assume that F_i and R depend continuously on their arguments. If the data α_j of equation (2) and their first derivatives are continuous and the initial guess U^0 is Lipschitz continuous on $\bar{\Omega}$, then for each k there are constants $C^k > 0$ and $\lambda^k > 0$*

$$\|u_j^{k+1} - u_j^k\|_T < C^k(e^{t\lambda^k} - 1)\|u_j^k - u_j^{k-1}\|_T \tag{6}$$

when $0 < t < T$ and for $1 \leq j \leq m$ and finite $k \geq 1$.

First, the boundedness and continuity of the iterates is established.

LEMMA 2. *Assume that F_i and R depend continuously on their arguments. Suppose the data α_j of the equation (2) and their first derivatives are continuous, and the iterate U^k is Lipschitz continuous on $\bar{\Omega}$, then U^{k+1} is Lipschitz continuous for $0 < t < T$.*

To establish this lemma, the equation governing the iterates will be simplified. Then, the iterate will be shown to be bounded, and the continuity of the iterates will be established.

PROOF. Equation (5) may be written

$$\frac{\partial u_j^{k+1}}{\partial t} + g_{j,j}^k u_j^{k+1} + \sum_{l=1,l\neq j}^{m} g_{l,j}^k u_l^k = 0, \tag{7}$$

where

$$g_{l,j}^k = r_{l,j}(t,x,U^k) + \sum_{i=1}^{n} f_{i,l,j}(t,x,U^k)\frac{\partial}{\partial x_i}.$$

The Lipschitz continuity of U^k and the continuity of the boundary data results in the coefficient $g_{j,j}^k$ and the summation term in the equation being bounded. Since the domain is bounded, iterate U^{k+1} will also be bounded.

An equation governing $\partial u_j^{k+1}/\partial x_i$ may be derived by taking the partial derivative of equation (7) with respect to x_i to obtain

$$\frac{\partial}{\partial t}\left(\frac{\partial u_j^{k+1}}{\partial x_i}\right) + g_{j,j}^k\left(\frac{\partial u_j^{k+1}}{\partial x_i}\right) + S = 0, \tag{8}$$

where the source term S and the coefficient $g_{j,j}^k$ of this equation are functions of u_j^{k+1} and the values and first derivatives of R, F_i, and U^k. The boundedness of u_j^{k+1} has been established, and the boundedness of the remaining terms follow from the assumptions made for this lemma. Thus, the boundedness of S and $g_{j,j}^k$ follow. From the boundedness of these terms, the boundedness of $\partial u_j^{k+1}/\partial x_i$ (and hence the Lipschitz continuity of u_j^{k+1}) follow. ∎

To establish Theorem 1, an equation governing the difference between iterates is derived. Since the iterates are Lipschitz continuous and the domain is bounded, appropriate upper and lower bounds for U^{k+1} will be determined. Then the difference is shown to be bounded with solutions of ordinary differential equations.

PROOF (OF THEOREM 1). The components of the difference $Z^k = (z_1^k, \ldots, z_m^k)$, where $z_j^k = u_j^{k+1} - u_j^k$ are governed by the equation

$$\frac{\partial z_j^k}{\partial t} + \sum_{l\neq j}[g_{l,j}^k z_l^{k-1} + (g_{l,j}^k - g_{l,j}^{k-1})u_l^{k-1}] + g_{j,j}^k z_j^k + (g_{j,j}^k - g_{j,j}^{k-1})u_j^k = 0. \tag{9}$$

The initial data is $z_j^k(0,x) = 0$. By the Mean Value Theorem

$$g_{l,j}^k - g_{l,j}^{k-1} = Z^{k-1} \bullet G_{l,j}^k,$$

where

$$G_{l,j}^k = \nabla_u r_{l,j}(t,x,\hat{U}) + \sum_{i=1}^{n} \nabla_u f_{i,l,j}(t,x,\hat{U})\frac{\partial}{\partial x_i}.$$

Here, $\hat{U}$ is some function bounded above and below by $\max(U^{k-1}, U^k)$ and $\min(U^{k-1}, U^k)$, respectively. Thus, equation (9) may be written as

$$\frac{\partial z_j^k}{\partial t} + g_{j,j}^k z_j^k = -Z^{k-1} \bullet \left[G_{j,j}^k u_j^k + \sum_{l\neq j} G_{l,j}^k u_l^{k-1}\right] - \sum_{l\neq j} g_{l,j}^k z_l^{k-1}. \tag{10}$$

Using the lemma, the absolute value of the terms on the right hand side of equation (10) are bounded by $C_j^k\|Z^{k-1}\|$, for some constant C_j^k. In adition, the coefficient of z_j^k is bounded above (below) by some constant $\bar{\lambda}^k$ ($\underline{\lambda}^k$). Thus, the functions

$$\overline{\omega_j} = tC_j^k\|Z^{k-1}\|_T e^{\overline{\lambda^k} t}$$

and

$$\underline{\omega_j} = tC_j^k\|Z^{k-1}\|_T e^{\underline{\lambda^k} t}$$

are upper and lower bounds, respectively, for $\|z_j^k\|$. With $C^k = \max_j C_j^k$ and $\lambda^k = \max_j \lambda_j^k$, the theorem is established. ∎

Theorem 1 did not establish the boundedness of C^k and λ^k as k goes to infinity. With the assumption that these limits are bounded below, the convergence of the iteration follows immediately from the theorem.

COROLLARY 3. *Assume that F_i and R depend continuously on their arguments. Also assume that the data α_j of the equation (2) and their first derivatives are continuous and the initial guess U^0 is Lipschitz continuous on $\bar{\Omega}$. If there are bounds $C > C^k$ and $\lambda > \lambda^k$ for all $k \geq 1$, then the iteration defined by (5) converges to a solution of (1).*

PROOF. From Theorem 1,

$$\|u_j^{k+1} - u_j^k\|_T < C(e^{t\lambda} - 1)\|u_j^k - u_j^{k-1}\|_T.$$

It is possible to choose a temporal bound $\hat{T}$ such that $C(e^{t\lambda} - 1) < 1$ when $T < \hat{T}$, and the iteration is a contraction.

Since the iteration is a contraction, it must converge to a fixed point $U^\infty = U^k = U^{k+1}$. Putting U^∞ into equation (5), clearly $U = U^\infty$ is a solution to (1)–(2). ∎

Thus, there are reasonable conditions under which the iteration will converge. Stronger results would likely be possible, especially considering that the experiments presented later in the paper do not satisfy all of the conditions of the analytic results.

5. ALGORITHM

An algorithm based on the iteration is presented here. The algorithm is a combination of the iteration presented in Section 3 with a temporal partitioning and a stopping criteria.

The temporal variable may need to be partitioned into several regions. For example, it is possible for the method to diverge if T is too large. In addition, the iteration requires that the solution be stored for the entire temporal region, possibly requiring too much memory. These problems are resolved by partitioning time into Q sections $0 < T_0 < T_1 < \cdots < T_Q = T$. The iteration will be performed on the partition $\Omega_q = (T_{q-1}, T_q] \times \Pi$ of the domain, using the solution at time $t = T_{q-1}$ from the iteration on Ω_{q-1} as the initial condition.

The stopping criteria is based on the norm of the difference between iterates as the time T_q. When the norm is less than some user specified tolerance, then the iteration is assumed to have converged.

The algorithm is independent of the discretization used in Step II.C.1 and of the initial guess used in Step II.A.1. Since the coefficients have nonlinear behavior, discretizations should be chosen that are appropriate for nonlinear problems. Notice that if an explicit discretization is used, then each temporal partition should contain at least two time steps. When a single time step is used with an explicit discretization, then the algorithm is no different than applying the explicit discretization directly to the linearization of equation (1).

I. Initialize.
   A. Set temporal partition counter to $q = 1$.
   B. Apply initial data (2) to the solution.
II. Determine the solution on temporal partition q.
   A. Doall $j = 1$ to n (j is the index for the elements of the solution).
      1. Determine initial guess u_j^0 for $(T_{q-1}, T_q) \times \Pi$.
   B. Initialize iteration counter $k = 1$.
   C. Doall $j = 1$ to n (do one iteration).
      1. Solve equation (5) to obtain u_j^k then
      2. Compute the norm of the difference $\kappa_j^k = ||u_j^k - u_j^{k-1}||$.
   D. If $K^k = \sum_j \kappa_j^k > TOL$
      1. Then increment k and go to Step C.
      2. Else finished with this temporal partition.
         a. Restart by using the solution at $t = T_{q-1}$ as initial conditions.
         b. Increment q and goto Step II.

Algorithm 1. Iteration with Restart.

Note the simplicity of the algorithm. There is little complexity added beyond implementation of the discretizations. This makes the method attractive when trying to obtain a solution with a minimal programming effort. In addition, there is potential to exploit parallelism on several levels. Large-grain parallelism is available through the decoupling of the equations, and may be further enhanced by using domain decomposition techniques (see [7]). Smaller-grain parallelism may be exploited by using appropriate data structures for each of the subdomains [8,9], and by choosing an appropriate numerical scheme to solve the PDEs. Since the number of iterations increases with the size of the temporal partition, there is a tradeoff between the cost of communication and synchronization and the number of iterations. Larger temporal partitions will result in larger pieces of work that can be done independently, but more work will be required. Exploitation of parallelism is an area of further research.

6. NUMERICAL EXPERIMENTS

In this section, Algorithm 1 is used to solve a system of nonlinear hyperbolic equations. These numerical experiments are a demonstration of the convergence of the algorithm. The implementation of the algorithm has not been optimized or parallelized; thus only the convergence results are presented. Experiments also examine the effects of the size of the temporal partition on the number of iterations. All the experiments were performed on an Ardent Titan, and required less than 5 minutes execution time for each run.

The equations solved are a two-dimensional form of Burger's equation

$$u_t + uu_x + vu_y = 0 \tag{11}$$

$$v_t + uv_x + vu_y = 0, \tag{12}$$

where time has the range $0 < t < 1$ and the spatial domain is the unit cube $\Pi = \{(x, y)|0 > \Delta(x, y) = t(\max[(x - 1/2)^2, (y - 1/2)^2] - 1/4\}$. The equations are capable of simulation of the physical phenomena that arise in computational fluid dynamics. Namely, with initial and boundary data

$$u = 2.0(.5 - y),$$

$$v = .1 + .9x,$$

equations (11)–(12) will be used to model shocks. The data are applied for $t = 0$ and for the inflow portion of Π which is defined by $y = 0$, $y = 1$ and $x = 1$. The solution is smooth initially, and develops a shock (see Figures 1–2). Thus, the examples satisfy all of the conditions of the theoretical results for the initial partition Ω_1, and evolve into a nonsmooth function that violates

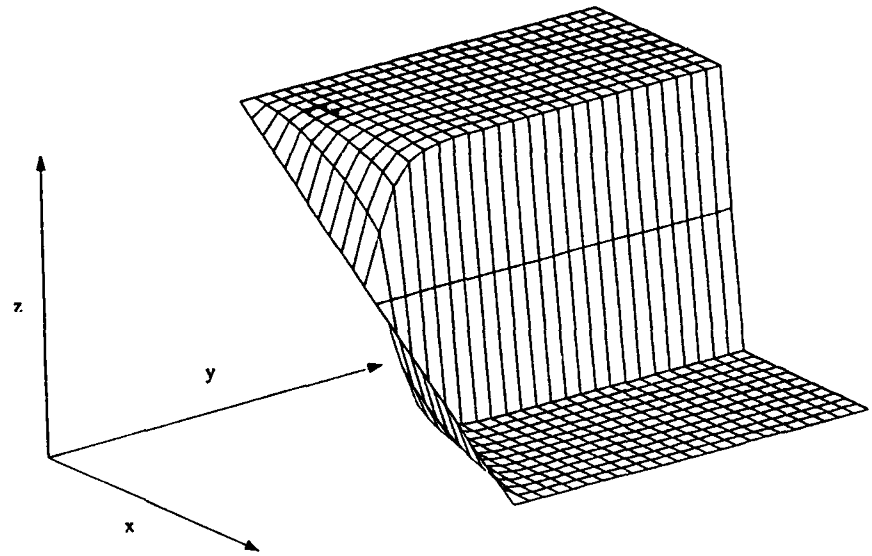

Figure 1. U at $t = 0.8$.

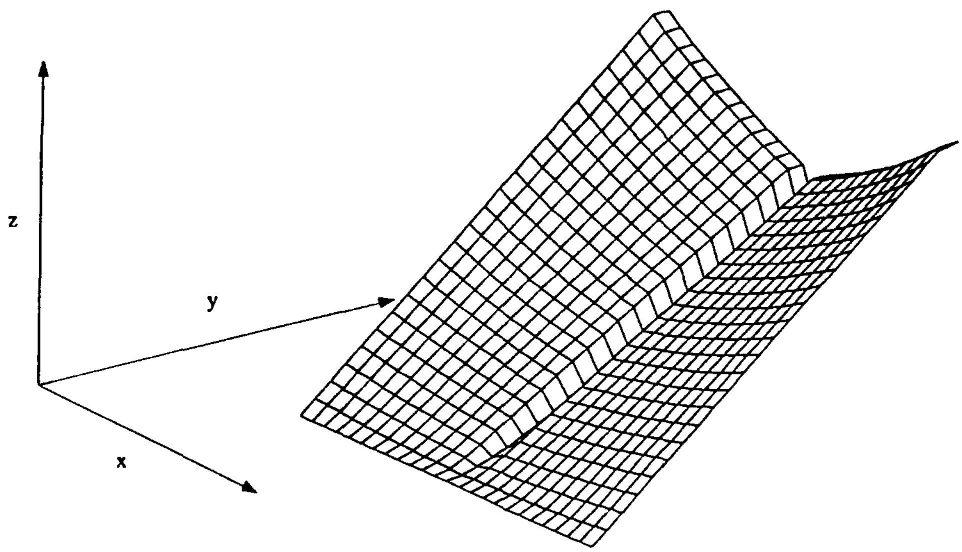

Figure 2. V at $t = 0.8$.

these conditions. Thus, the experimental results will demonstrate that the algorithm works even when some of the conditions used to establish the theory are violated.

The experiments were performed using two different explicit discretizations, a second-order MacCormack scheme [3], and a first-order upwind scheme. The upwind scheme consists of applying a backward or forward difference based on the sign of the coefficient of the spatial derivative being differenced.

To improve computational efficiency, the discrete analog of the L_1 norm at time $t = T_q$ is used in place of the L_2 norm in Step II.C.2. Thus, κ_j^k is

$$\kappa_j^k = \frac{1}{P} \sum_{p=1}^{P} |u_{j,p}^k - u_{j,p}^{k-1}|, \tag{13}$$

where P is the number of spatial grid points, and $u_{j,p}^k$ is the discrete value of component j of iterate k. Including boundary points, there were 25 unknowns in each direction, thus the total number of points was $P = 625$. The time step was $\Delta t = .001$.

The initial guess on Ω_q is

$$u^0(t,x,y) = u^{\hat{k}}(T_{q-1},x,y)$$

and

$$v^0(t,x,y) = v^{\hat{k}}(T_{q-1},x,y),$$

where $(u^{\hat{k}}, v^{\hat{k}})$ was the last iterate for partition Ω_{q-1}. This resulted in a good initial guess as measured by the norm K^1 of the difference between the initial guess and the first iterate (see Table 1).

Table 1. Average Number of Iterations.

Number of Time Steps	Upwind		MacCormack	
	K^1	Num Iter	K^1	Num Iter
20	9.94×10^{-3}	3.8	2.93×10^{-2}	4.1
50	2.43×10^{-2}	4.7	7.12×10^{-2}	6.7
100	4.69×10^{-2}	5.6	1.38×10^{-1}	8.6

The results in Table 1 show the effects of varying the temporal partition size on the convergence of the method with the initial guess as described above. The data are the average over the total number of temporal partitions. Thus, corresponding to the number of time steps of 20, 50 and 100 are partition sizes $T_q - T_{q-1} = .04$, .1, and .2, respectively. The results in the table are the data averaged over all of the temporal partitions required to take the 500 time steps necessary to obtain the solution at $T = 1$. Hence, the numbers are averaged over 50, 10 and 5 values for the number of time steps of 20, 50 and 100 respectively. Experiments using a constant initial guess confirm that the additional iterations required for the larger partitions are not due to the larger beginning norm K^1. Further work in this area is necessary to establish the reason for this trend; however, the larger computational domain resulting from the larger temporal partition is a likely cause.

The data from numerical experiments in Table 2 demonstrate linear convergence. A steady shock has formed in the solution by time $t = .8$; thus, the data in the table reflect that the reduction in the difference was still linear when the solution has large gradients that violate the conditions of the theorems. A result that is not reflected in the table is the monotonic reduction in the error for each component of the solution in each iteration, as predicted by the analysis.

Table 2. Reduction in Norm for all Iterations.

Iteration	Num Stopped	Ave Reduction
2	0	6.99×10^{-2}
3	3	5.48×10^{-2}
4	1	5.87×10^{-2}
5	3	4.09×10^{-2}
6	3	2.88×10^{-2}

The tolerance TOL for the stopping criteria was set to 2×10^{-7}, slightly lower than machine precision. The iteration on each partition used the TOL stopping criteria. This resulted iterations on various temporal partitionings stopping at different iterates. The number of iterates on a partition that stopped at the iteration number in the left column is reported in the 'Num Stopped' column. The average reduction column is thus the relative reduction K^k/K^{k-1} as measured over the number of times the iteration made it that far.

The experiments confirm and go beyond the analytic results. The monotonic reduction in the difference between iterates of Theorem 1 was observed. The experiments went beyond the analytic results by showing a linear reduction of slightly more than a digit of accuracy on a model problem with a shock. Larger temporal partitions required more iterations. This indicates that the larger partitions will require more computations, but may require fewer synchronizations. To reach any conclusions, this issue could be studied once the algorithm has been implemented in a parallel processing environment.

7. CONCLUSIONS

An iterative algorithm for the efficient solution of systems of nonlinear hyperbolic equations has been discussed. The method is simple to implement, and has parallelism that can be exploited on several levels. In this paper, convergence was established analytically for continuous solutions. Numerical experiments demonstrated the monotone convergence predicted by the analysis, both when the solution was smooth, and when the problem had shocks resulting from the nonlinearities.

REFERENCES

1. J.S. Scroggs, The solution of a parabolic partial differential equation via domain decomposition: The synthesis of asymptotic and numerical analysis, PhD thesis, University of Illinois at U-C, (1988).
2. J.S. Scroggs, A physically motivated domain decomposition for singularly perturbed equations, *SIAM Journal on Numerical Analysis* **28** (1) (1991).
3. D.A. Anderson, J.C. Tannehill and R.H. Pletcher, *Computation Fluid Mechanics and Heat Transfer*, Hemisphere Publishing Corp., McGraw-Hill, New York, (1984).
4. D.L. Bernstein, *Existence Theorems in Partial Differential Equations*, Princeton University Press, Princeton, New Jersey, (1950).
5. R. Courant and D. Hilbert, *Methods of Mathematical Physics, Volumes I and II*, Interscience, New York, (1962).
6. G.S. Ladde, V. Lakshmikantham and A.S. Vatsala, *Monotone Iterative Techniques for Nonlinear Differential Equations*, Pitman Publishing, Inc., Marshfield, Massachusetts, (1985).
7. R. Glowinski, G.H. Golub, G.A. Meurant and J. Periaux, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, SIAM, Philadelphia, PA, (1988).
8. D. Gannon and J. van Rosendale, On the structure of parallelism in a highly concurrent PDE solver, *J. Par. Dis. Comp.* **3**, 106–135 (1986).
9. W.D. Gropp, Local uniform mesh refinement with moving grids, *SIAM J. Sci. Stat. Comp.* **8** (3), 292–304 (1987).